REMARKS/ARGUMENTS

A. The Office Action rejected claims 1-22 under 35 USC 103(a) as being unpatentable over Ichbiah in view of Lu. Applicant respectfully traverses the rejection.

The Examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness.[1] If the Examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of non-obviousness.[2]

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure.[3]

Applicant respectfully traverses the § 103 rejection because the office action has not established a *prima facie* case of obviousness.

The references do not teach or suggest all the claim limitations.

As to claim 1, Lu does not disclose storing in a memory a first data structure encoding a plurality of words and corresponding abbreviations followed by selecting a word in the text to be converted to an abbreviation and converting the word to an abbreviation using the first data structure.

[1] MPEP Sec. 2142.

[2] Id.

[3] Id. (emphasis supplied)

As described at Col. 1 line 49 to Col. 2 line 2, Lu converts Long Case Names (LCNs) to Short Case Names (SCNs) through the following steps:

1) storing a batch of LCNs in memory;

2) converting each work in an LCN into a low-level token;

3) creating high-level tokens to represent and classify groups of low-level tokens;

4) using the identities of the high-level tokens, along with SCN creation rules, to determine which of the low-level tokens represent text that should be included in the SCN; and

5) converting selected low-level tokens into SCNs.

It is only if there is any ambiguity as a result of this process that the system sends LCNs for which multiple SCNs have been generated to human editors for selection of the correct SCN.

That is, the words (LCNs) have already been converted to abbreviations (SCNs) using tokenizing and parsing rules, not a data structure of words and corresponding abbreviations, before, in the exceptional case, a list of LCNs and corresponding SCNs is displayed to the editor. Thus, the Lu process does not: 1) store a data structure in memory that encodes a list of words and corresponding abbreviations and then 2) use this data structure to convert a word to a corresponding abbreviation. The step of displaying the LCNs and SCNs to the user does not "convert the word to the abbreviation"—this has already been done without using any data structure.

Claim 1 is therefore allowable.

Claims 9-10 contain additional elements or limitations beyond allowable claim 1 and are also allowable.

Regarding claims 3-6, Ichbiah does not select a word in the text to be converted to an abbreviation using a keyboard or a mouse. Col. 3 lines 63-65 only teaches displaying a list of words that have already been converted from abbreviations to the user for selection, in the event that more than one word corresponds to a given abbreviation. Ichbiah only converts abbreviations to words, not vice-versa, and therefore does not teach converting words to abbreviations.

Regarding claims 7-8 and 11-12:

Amended claim 7 is dependent upon claim 1 and therefore includes all limitations of claim 1. As discussed above, Lu does not disclose converting a word to an abbreviation using a first data structure. Lu therefore also cannot scan the text to convert words to abbreviations using a first data structure.

Claim 11 is dependent upon allowable claim 7 and is therefore also allowable. Claims 8 and 12 are dependent upon allowable claim 1 and are therefore also allowable.

Regarding claim 13, at page 3 of the Office Action, the Examiner states "Ichbiah does not explicitly disclose storing in memory a first data [structure] encoding a plurality of words and corresponding abbreviations." Therefore, the statement at page 6, line 17-18 of the Office Action is self-contradictory ("Ichbiah discloses selecting an abbreviation from the first data structure (abstract; col. 3 lines 50-65)..."). In any case, this is not the teaching of col. 3 lines 50-65. Rather, the teaching is of choosing one of a set of matching words generated from an abbreviation, not selecting the abbreviation and inserting the abbreviation into the text at a position selected by the user.

Claim 14 is dependent upon allowable claim 1 and is also therefore allowable.

Claim 15 is allowable for the reasons given above in regard to claim 1. In addition, there is no disclosure in the references of "the user instructing the data processing method to select a position in the text for insertion of an abbreviation." The only reference that converts words to abbreviations, Lu, does not teach this ability. Lu is intended to run as a fully automatic method of simply converting (one way) Long Case Names into Short Case Names. There is no disclosure of interaction with the user to select a position in the text for insertion of an abbreviation.

Claim 17 is allowable for the reasons given above. Claims 18-22 contain additional elements or limitations beyond allowable claim 17 and are therefore also allowable.

For the above reasons, Applicant respectfully requests the allowance of all claims and the issuance of a Notice of Allowance.

Respectfully submitted,

Dated: 12/15/03

By Nelson R Capes
Nelson R. Capes (Reg. No. 37,106)
BRIGGS AND MORGAN
2200 IDS Center
80 South Eighth Street
Minneapolis, MN 55402
Telephone: (612) 977-8486